Final Version

*** Please do not distribute ***

High Speed Electrical Signalling: Overview and Limitations

Mark Horowitz, Chih-Kong Ken Yang and Stefanos Sidiropoulos
Computer Systems Laboratory
Stanford University
Stanford, CA 94305

Correspondence Address:
Mark Horowitz
Gates Computer Science 3A
Room 306, M/C 9030
Stanford University
Stanford, CA 94305-9030
Phone: (415) 725-3707
FAX: (415) 725-6949

Abstract — **Improving fabrication technology enables not only the scaling of on-chip gate speeds but also the data rate of inter-chip communication interfaces. Simple low latency off-chip interfaces are limited by the maximum clock frequency that can be propagated on-chip. More complex serial links break this barrier, by employing high fan-in multiplexing transmitters and high fan-out de-multiplexing receivers, thus achieving bit-times on the order of an on-chip gate-delay. As technology scales, the higher signalling rates exceed the bandwidth of the external communication media. Fortunately, the increasing integration levels also enable the use of complex modulation and coding schemes to better utilize the available signal power and bandwidth for higher data rates. However, the complexity of these schemes increases the power, area, and latency overhead of the link, thus limiting their application to bandwidth-critical and wire-limited systems. With smaller feature-sizes and higher resolution requirements, the increased effect of on-chip mismatches limits both the data and clock recovery performance, posing future circuit design challenges.**

Keywords — **Communication system interfaces, Signalling, Synchronization, Clock and data recovery, PLL's**

High Speed Electrical Signalling: Overview and Limitations

Mark Horowitz, Chih-Kong Ken Yang, and Stefanos Sidiropoulos
Computer Systems Laboratory
Stanford University
Stanford, CA 94305

I. Introduction

Advances in IC fabrication technology coupled with aggressive circuit design have led to an exponential growth of the speed and integration levels of digital IC's. In order for these improvements to truly benefit the overall system performance, the communication bandwidth between systems and IC's must scale accordingly. Currently, communication links used in a variety of applications are all approaching Gbit/sec data rates. The applications of these links are diverse: computer-to-peripheral connections [1], local area networks [2], memory-busses [3], and multiprocessor interconnection networks [4], [5]. There is concern that these links will soon run into the fundamental limits of electrical signalling. To examine this concern, we look at the limitations of CMOS implementations of high-speed links and show that the performance of the CMOS links should continue to scale with technology. However to handle the finite bandwidth of the interconnects, more sophisticated signalling methods will soon be needed.

Although CMOS circuits are typically slower than circuits implemented in non-mainstream technologies, their lower cost makes their use much more attractive especially for ubiquitous applications. The overall system cost becomes even lower when the signalling components are implemented as macro-cells that can be integrated on the same die with a microprocessor or signal processing block. For this reason, traditional bipolar or GaAs Gbit/sec links are not addressed in this paper even though their limitations are similar in nature.

II. Issues in Signalling

The components of a signalling system, shown in Figure 1, are the transmitter, the channel, and the receiver. The transmitter converts digital information to a signal/waveform on the transmission medium. This medium on which the signal travels, e.g. board trace, coaxial cable or twisted pair, is commonly called the communication channel. The receiver on the other end of the channel restores the signal to the original digital information by sampling and quantizing the signal. Tightly coupled to the signal transmission and reception is the clock generation and timing

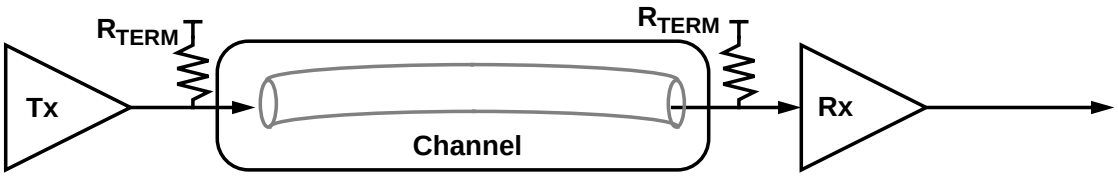


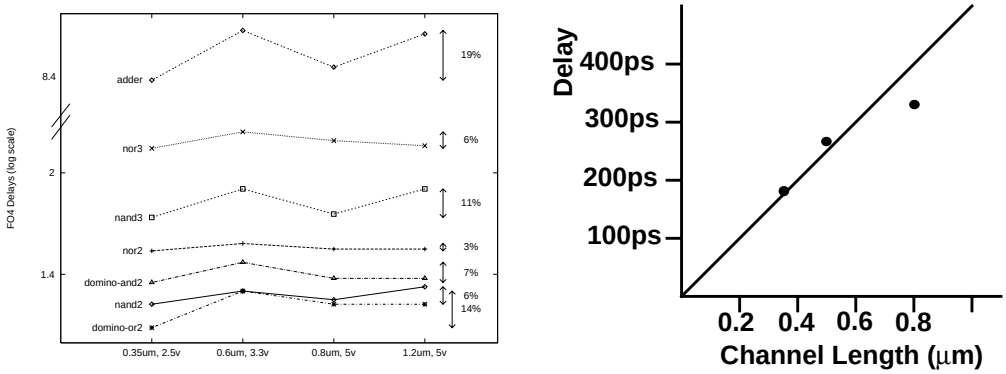
Figure 1: Signalling system components

recovery. The timing recovery, often embedded in the receive-side, adjusts the phase of the clock that strobes the receiver at the optimal position to sample the signal on the channel.

In order to discuss the performance limitations of these link components, we need a way to measure a CMOS circuit's performance that indicates how the performance will scale with technology. Such a metric is described next, and is followed by a description of the metrics that are used to characterize a link's performance. Section III and Section IV then use these metrics to evaluate the signalling and clocking circuits used in links. Two types of links are described: a parallel link optimized for low latency, and a serial link optimized for the highest possible data rate. As signalling rates and/or cable length increase, the attenuation of the channel begins to limit performance by reducing the available bandwidth. So, Section VI discusses techniques such as channel equalization and coding that deal with channel bandwidth limitations. The maximum theoretical data rate that can be achieved is governed by Shannon's channel capacity theorem, but the resolution of high-speed digital-to-analog and analog-to-digital converters will limit the practical data rates.

A. CMOS Technology Performance Metrics

As technology is scaled, the speed of the basic circuits improves. Fortunately, the delay of all CMOS circuits scales roughly the same way. Thus the ratio of a circuit's delay to a reference circuit remains about the same. We take advantage of this fact by creating a metric called a fanout of four (FO-4) delay. A "FO-4 delay" is the delay of one stage in a chain of inverters, where each of the inverters in the chain drives a capacitive load (fanout) that is 4x larger than its input



(a) (b)

Figure 2: Fan-out-4 metric performance

capacitance. Figure 2-(a) illustrates the normalized delay of various circuit structures versus technology and voltage scaling, demonstrating a worst case prediction accuracy of 20% for a relatively complex circuit structure. Figure 2-(b) shows the actual FO-4 inverter delay for various technologies. Interestingly, the FO-4 delay for these processes can be roughly approximated by 500-ps/μm (gate length). In a 0.5-μm technology a 1-Gbps data stream has bits that are 4 FO-4 long, while at 0.25-μm link circuit has 8 FO-4 delay per bit.

B. Link Performance Metrics

The performance of a link is pre-dominantly characterized by its data bandwidth. In many applications latency, power and die area are also critical issues. These latter issues are generally more important in intra-system parallel links since the cost is multiplied by the number of wires. Data bandwidth or bit-rate is often confused with symbol-rate, but symbols may contain multiple bits of information. For example, a binary NRZ (non-return-to-zero) signal has the same symbol-rate as bit-rate. In contrast, a 4-level pulse-amplitude-modulated (PAM) signal (comprising 2-bits per symbol) has a bit-rate twice that of its symbol-rate.

Another important link performance metric, the bit error rate (BER), is a measure of how many bit errors are made per second. BER is important not only because it reduces the effective system bandwidth, but because in many systems, application of error correction techniques can prohibitively increase the system cost. The errors are caused by noise and imperfections in the system. The intrinsic sources of noise are the random fluctuations due to the inherent thermal and shot noise of the passive and active system components. However, especially in VLSI applications, other non-fundamental sources of noise limit the link performance. These noise sources include coupling from other channels, switching activity from other circuits integrated with the link circuitry, and reflections induced from channel imperfections. These types of noise typically have a non-white frequency spectrum, and exhibit strong data dependencies. Moreover, their overall power is often proportional to the power of the transmitted signals.

In the sections that follow we examine circuits optimized for two different applications: short parallel links and medium-long serial links. In medium-long serial links, the goal is as much bandwidth as possible, because the wire is the critical resource. Since there is only one transmitter and receiver, the area and power costs are not paramount. In addition, the latency of the link's active circuits is not a large concern, because the overall system latency is already dominated by the channel delay. To get maximum performance, the link is operated until the BER is on the order of 10^{-9}-10^{-11} and an error correcting code is used to increase the system robustness. Short parallel links are quite different. Intra-system interconnects (e.g. MP networks, CPU-memory links), typically exhibit much lower channel delays. Hence, the impact of the incremental latency of the link circuitry on the overall system performance is much more important. Furthermore, the large number of transmitter and receiver circuits (tens to hundreds) requires modest area and

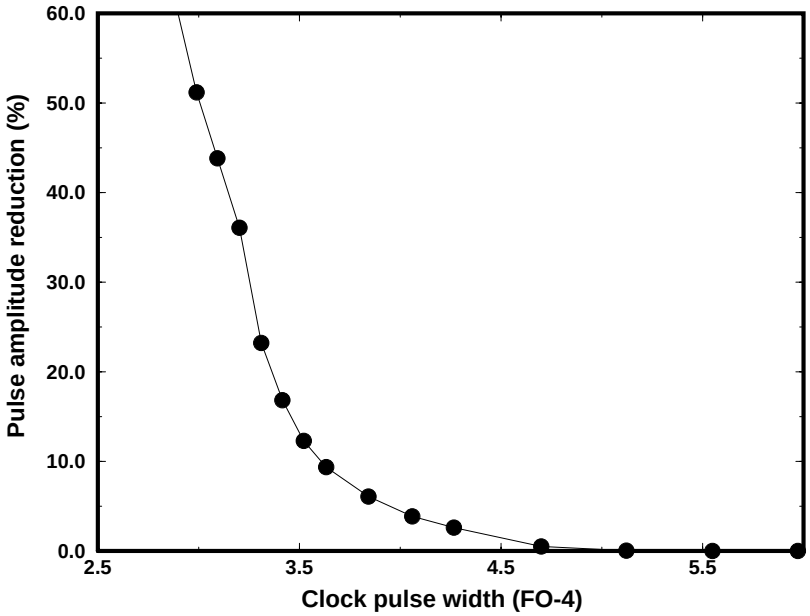


Figure 3: Pulse amplitude reduction as a percentage of pulse width

power for each circuit. To further reduce latency and complexity, these systems target aggregate BER's that are practically zero (or much less than the projected mean-time-between-failures of the overall system) so no error correction is needed.

III. Signalling Circuits

A natural limitation of a link's bandwidth is the on-chip data rate, which is dictated by the speed of the logic that processes the transmitted/received data, and the speed of the clock. A clock must be buffered and distributed across the chip. Figure 3 shows the effect of driving different pulse widths (expressed in the FO-4 delay metric) through a clock buffer chain with a fanout of four per stage. As the pulse-width reduces, the amplitude of the clock pulses becomes significantly reduced. The minimum pulse that makes it through the clock chain is around 3 FO-4 delays. Since a clock is actually a rising pulse followed by a falling pulse, the minimum clock cycle time will be 6 FO-4. Accounting for margins, it is unlikely to see clocks faster than 8 FO-4 cycle time. With one bit per cycle, this would give only 500-Mb/s in a 0.5-μm technology. One simple way to overcome this limitation is by utilizing parallelism, which allows the on-chip processing circuits to operate at a lower frequency than the off-chip data rate. The following two section look at how parallelism is used in transmitter and receiver design.

A. Transmitter Design

Figure 4, shows the block diagram of a transmitter implementing parallelism through multiplexing the on-chip parallel data into a single serial bit-stream. The primary bandwidth limitation in this system stems from either the multiplexor or the clocks. When the bit-times are

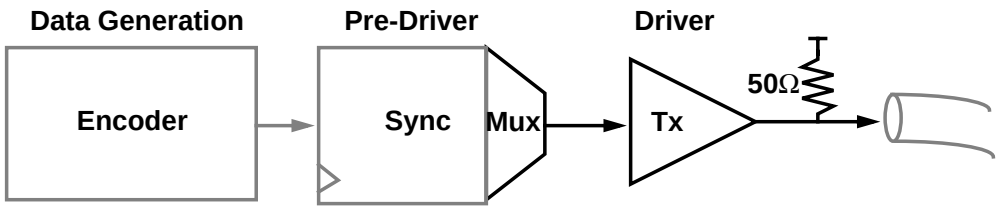


Figure 4: Multiplexing pre-driver

long, the multiplexor output achieves full CMOS swing. This clipping of the multiplexor output voltage creates an inherent memory-less medium, i.e. the previous bit values do not affect the waveform of the bit currently being transmitted. However, when the bit-time is shorter than the multiplexor settling time, the output no longer swings fully, and the values of the previously transmitted bits affect the waveform of the current bit. This interference is called inter-symbol interference (ISI), and reduces both the timing and voltage margins of the transmitted signal. Figure 5 shows the effect of bit-time on the pulse width of the multiplexor output signal, assuming that the clock driving the 2:1 multiplexor is generated by a FO-2 buffer chain. While the multiplexor is fast enough for a bit-time of 2 FO-4's, a less aggressive clock buffering of a fanout of 3-4 per stage would increase the bit-time to at least 3 FO-4's. A larger fanin multiplexor could get around the clock problem, but its larger capacitance would decrease its performance to around 3 FO-4 delays. This 2:1 multiplexor solution is simple and cheap, and is used in short parallel links.

Instead of relying on the bandwidth of the on-chip multiplexor, we can utilize the high bandwidth available at the chip output pins. Since the chip's output drives a low (25-50Ω) impedance, a high bandwidth is available even with larger output capacitance. Figure 6-(a) shows the basic idea behind this technique implemented on a pseudo-differential current-mode output driver. To achieve parallelism through multiplexing, additional driver legs are connected in parallel. Each driver leg is enabled sequentially by well-timed pulses of a width equal to a bit-time. Since in this scheme the current pulses at the output are completely generated by on-chip

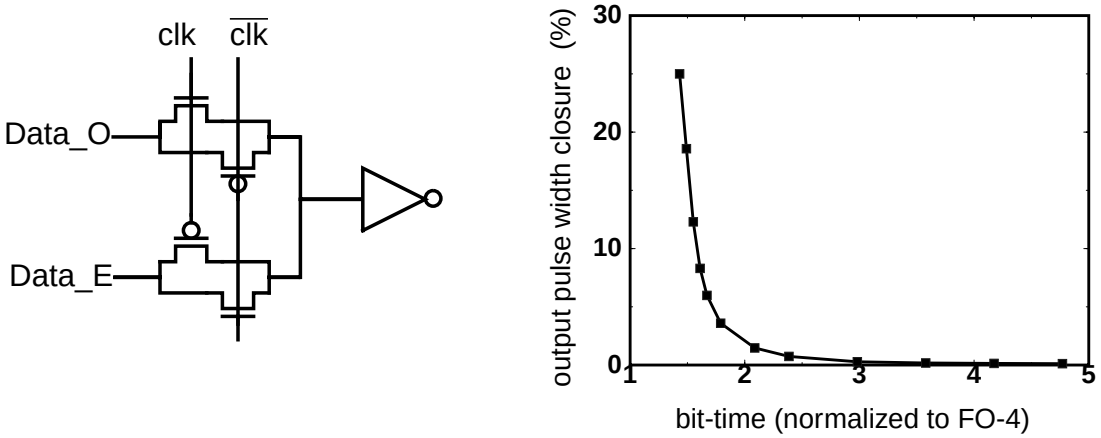


Figure 5: Effect of bit-time on output pulse width of a simple 2:1 multiplexor

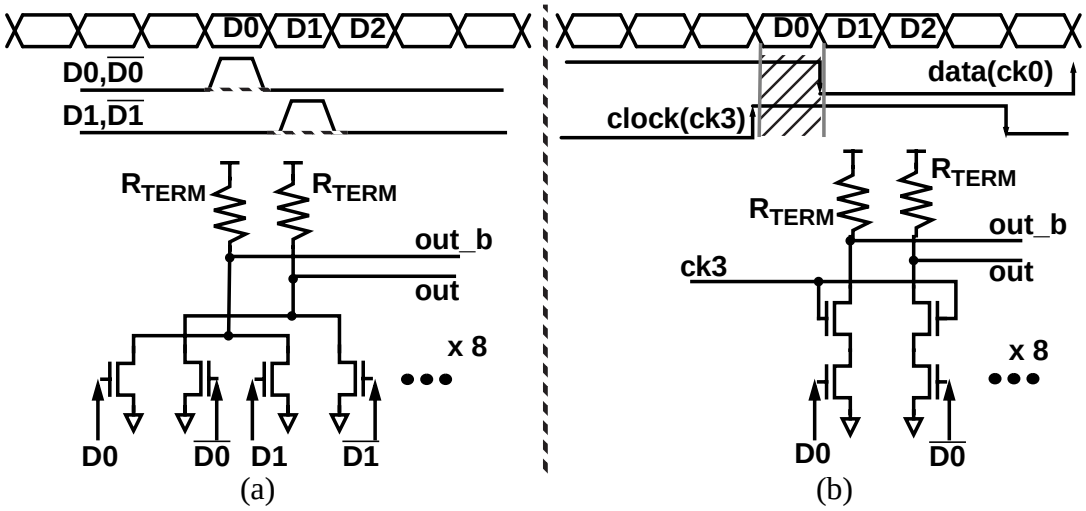


Figure 6: Higher bandwidth multiplexing at the output pin

voltage pulses, the minimum output pulse width is limited again by the minimum pulse width that can be generated on-chip [6]. An improved scheme, illustrated in Figure 6-(b), eases that limitation [7]. In this configuration two control signals, both at the slower on-chip clock rate, are used to generate the output current pulse. In this design, either the transition time of the pre-driver output or the output bandwidth determine the maximum data rate. Figure 7 illustrates the amount of pulse time closure with decreasing bit-width for an 8:1 multiplexor utilizing this technique. The minimum bit-time achievable for a given technology is less than a single FO-4 inverter delay. The cost of this scheme is a more complex clock source, since precise phase shifted clocks are required, and an increase in latency (measured in bit-times). The highest speed links use this technique, and achieve bit-times that are on the order of 1 FO-4.

B. Receiver Design

A more challenging problem exists in the design of the receiver front end. The typically small

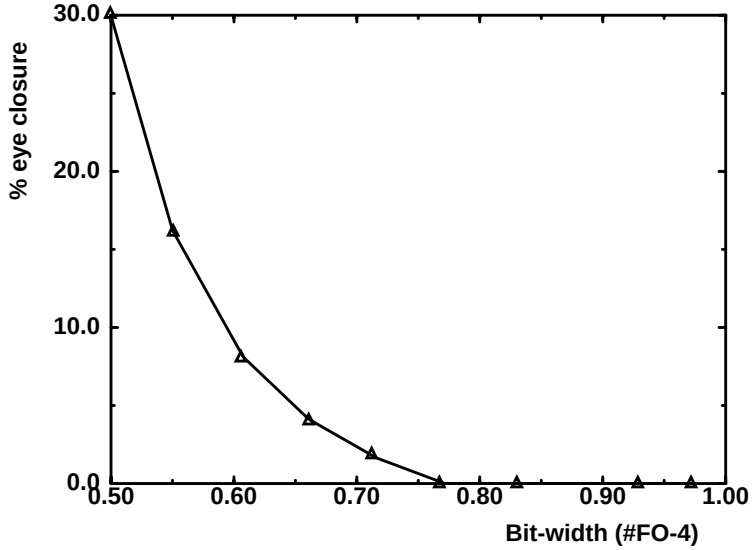


Figure 7: Effect of bit-time on pulse width of the higher bandwidth 8:1 multiplexor

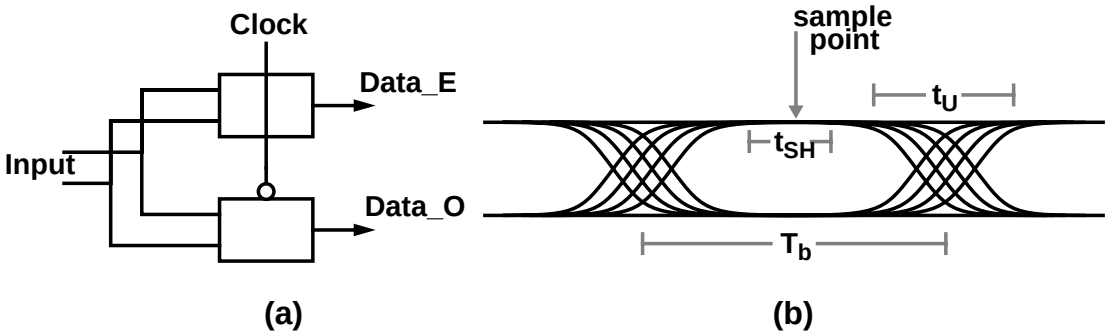


Figure 8: Simple 1:2 de-multiplexing and timing margin

swing signal on the channel must be restored to full CMOS swings. If a conventional amplifier is used to restore the signal, its output bandwidth would limit the achievable bit-rate causing inter-symbol interference. Once again, if parallelism is employed, the bandwidth requirements of any single receiver component will be relaxed increasing the maximum achievable data rate. This can be achieved by first sampling the data through multiple parallel samplers, each followed by an amplifier. Each amplifier now has a bandwidth requirement which is a fraction of the original single front-end amplifier bandwidth [8], [9]. To optimize the performance, a high gain-bandwidth-product amplifier is desirable. An effective design is a regenerative amplifier, whose gain is exponentially related to the bandwidth due to the positive feedback.

The most common form of this de-multiplexing is employed in low latency parallel systems [3], [10]. Two receivers are used on each input, one of them triggered by the positive and one by the negative edge of the clock as illustrated in Figure 8-(a). Each receiver has 1/2 cycle to sample (while resetting the amplifier) while another 1/2 cycle (one full bit-time) can be allocated for the regenerative amplifier to resolve the sampled value. The bit-rate achievable by this simple de-multiplexing structure is determined by the minimum operating cycle time of the receiver. Representative receiver designs can achieve bit widths on the order of 4 FO-4 delays, which match the clock limitations [3], [7], [10].

Similar to the transmitter, a higher degree of de-multiplexing can be employed by using well controlled clock spacing as shown in Figure 9. In this system, the de-multiplexing occurs at the input sampling switches [6],[7],[9]. The fundamental bandwidth limitation on this parallelized architecture is imposed by the ability to generate the finely spaced clock phases (discussed in Section IV) and the sampling aperture (or sampling bandwidth) of CMOS transistors. The sampling aperture is the amount of time required for the sampler to capture the input value. An NMOS transistor sampler can easily be sized for low enough resistance to have sampling aperture less than 1/3 FO-4. This performance is sufficient to robustly recover data with 1 FO-4 bit-times. However, a secondary and often more severe limitation is imposed by the input capacitance which depends on the sampling network design and the degree of de-multiplexing. In a well-balanced design, the maximum achievable data rate is simply the product of the inverse of the minimum cycle time of the front end receiver multiplied by the degree of de-multiplexing. The de-

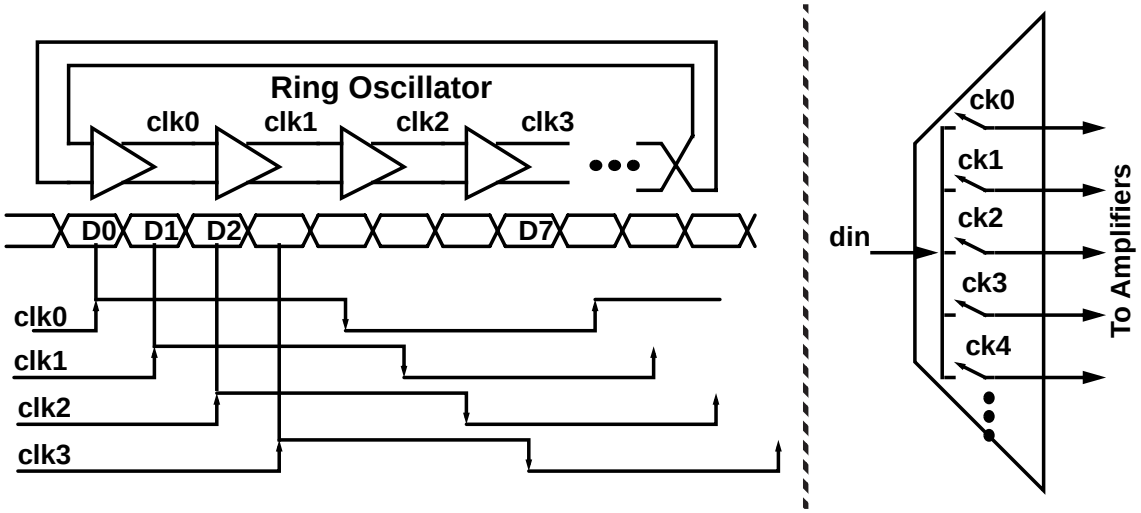


Figure 9: 1:8 de-multiplexing receiver

multiplexing degree is limited such that the input RC time constant is small enough to not cause additional ISI on the input signal.

Even when the sampling bandwidth is sufficient, performance can be degraded by sampling uncertainty (aperture uncertainty). For NRZ signalling, the sampling occurs in the center of the data bit for maximum timing margin. Figure 8-(b) illustrates that, because of uncertainty in the data, t_U, the sampling uncertainty cannot exceed t_{SU} otherwise errors occur. This sampling uncertainty is the sum of phase uncertainty in the sampling clock, static input offset of the amplifier, and sampling noise. Phase uncertainty, both static and dynamic, is the dominant source, and will be discussed in the following section

Another noise issue arises in many parallel links. Due to cost reasons these links typically utilize pseudo-differential signalling [3],[4],[14],[15]. In these systems, a reference voltage is shared among the receivers at each pin. This sharing of the reference voltage creates an imbalance on the load of the reference and input lines, causing on-chip power supply and substrate noise to be more heavily coupled on the shared reference line. This high frequency noise is manifested in the time domain in the form of reference voltage “spikes” that can be detrimental in the operation of a high-bandwidth sampling/regenerative receiver. In order to reduce these reference noise effects, the receiver should filter the high frequency reference noise. An effective filter is one that resembles the traditional communications integrate-and-dump filter [16]. Because the input signal in low latency parallel links is valid for more than the brief time instant of the sampling aperture, capacitors can be used to integrate current based on the input voltage difference (V_{in}-V_{ref}). At the end of the integrating period, the integrated voltage is resolved by a regenerative amplifier. Since input differential voltage “spikes” are averaged over the bit-time, the final polarity of the integrated signal is not affected, thus improving the overall system robustness [32].

Not surprisingly, transmitters and receivers have similar limitations. A simple 2-1

multiplexing/de-multiplexing transmitter/receiver pair can easily achieve bit-times of 3-4 FO-4 inverter delays and thus should continue to scale with technology. Improving data rates even further can be achieved by wider transmitter multiplexors and receiver de-multiplexors. By employing higher degrees of parallelism, these systems can achieve bit-times of approximately 1 FO-4 inverter delay. The main requirement on these systems is precise timing, placing more stringent requirements on the link synchronization circuits. The design of these circuits is the topic of the next section.

IV. Synchronization Circuits

To properly recover the bit sequence at the receiver end of the channel, the receiver's sampling clock phase needs to have a stable and pre-determined relationship to the phase of the incoming data, thus maximizing timing margins. This deterministic phase relationship becomes an even more stringent requirement in higher bandwidth systems. In these systems, the bit-rate is a multiple of the on-chip clock, requiring either an explicitly faster bit-clock, or multiple phases of lower frequency clocks with well-controlled phase relationship between them.

Clock quality can be characterized by phase offset and jitter. Phase offset is a static (DC) quantity which is equal to the difference between the ideal average position of a clock and the actual average position. This offset can refer both to the phase relationship between clock and data, as well as intra-phase offset in multiplexing systems with multiple clock phases. Jitter is the dynamic (AC) variation of phase and is dominated by on-chip power supply and substrate noise. Jitter is specified in terms of both short-term and long-term variations. Cycle-to-cycle jitter describes the short-term uncertainty on the period of a clock, while long-term jitter describes the uncertainty in the position of the clock with respect to the system clock source. In conventional digital design the most important requirement is minimizing cycle-to-cycle jitter. In high speed links, however, both quantities can be equally important. Low frequency jitter is caused by imperfections on the system clock source and slow temperature and operating voltage variations. This type of jitter can be tracked reasonably well by employing a phase locked loop. Medium frequency and cycle-to-cycle jitter are caused by on-chip supply and substrate noise and are the major concern.

A. Clock Generation Circuit Architectures

As mentioned above, a reliable and flexible method for dealing with the synchronization problem is to use on-chip active phase aligning circuits. Generally, these circuits fall in a class of control systems known as Phase-Locked Loops. These systems use negative feedback to align the phase of the on-chip receive or transmit clock to the phase of an external reference. For parallel links, this external reference is the clock distributed along with the data. For serial links, the reference is

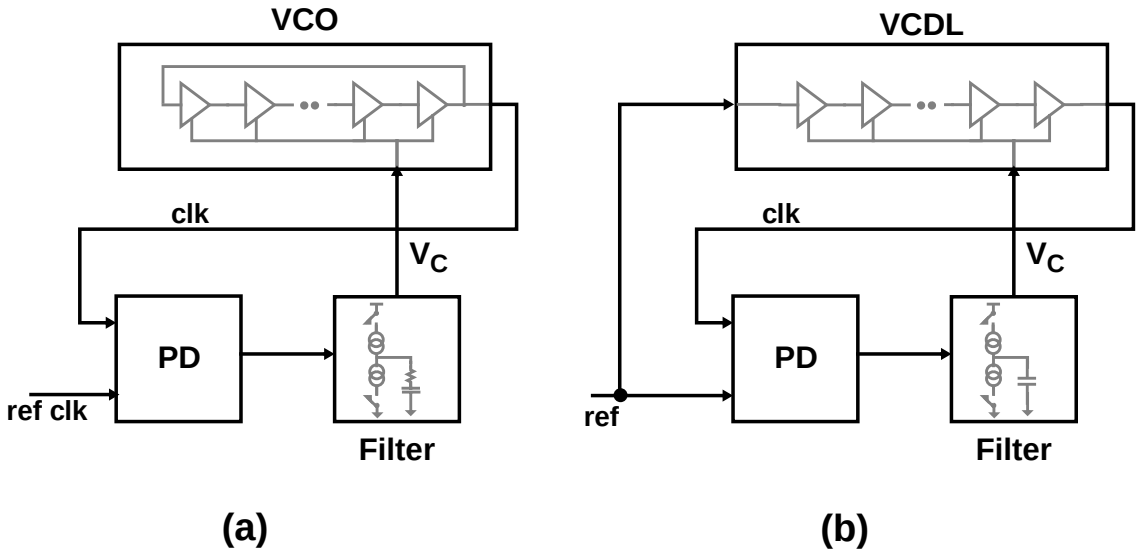


Figure 10: PLL and DLL loop topologies

more often extracted from the serial data stream. In a parallel link, the process of buffering this reference clock to drive multiple receivers alters the timing relationship between clock and data. By appropriately embedding the delay of the clock amplification/buffering in the feedback path, Phase-Locked Loops can cancel out this skew, thus fixing the phase relationship between the internal clock and external reference. Similarly, in a serial link, the timing information extracted from the data must be fed back to control the sampling phase. In order to truly improve the system timing margin, the additional fixed and time-varying phase uncertainty (i.e. offset and jitter) introduced by the phase aligning blocks must be minimized, which means minimizing the effect of supply and substrate noise.

Figure 10 shows two alternative control loop topologies that can be used in high speed signalling systems: VCO-based Phase-Locked Loops (PLL's), and delay-line-based phase-locked loops or Delay-Locked Loops (DLL's). The basic idea behind the operation of these two circuits is quite similar: they both try to drive the phase of their periodic output signal (*clk*) to have a fixed relationship with the phase of their input signal (*ref-clk*). A PLL employs a voltage controlled oscillator (VCO) to generate its output clock. The phase of that clock is compared with that of the reference by the phase detector. The output of the phase detector is filtered by the loop filter, generating the loop control voltage (V_C) which drives the control input of the VCO. Since a VCO integrates frequency to generate the phase of its output clock, a PLL is inherently a higher order control system. The transfer function of the system contains two poles at the origin: the first due to the phase integrating nature of the VCO, and the second due to the integrator usually embedded in the loop filter to achieve zero static-phase-error. To counteract the effect of these two poles, the loop transfer function must contain a stabilizing zero. This zero is usually implemented in the loop filter by employing a resistor in series with the integrating capacitor. This higher order nature of the PLL creates some design challenges. For example, the effects of process and environmental

conditions variations on the stabilizing zero position, might be detrimental on the loop stability [17], [18], [19]. On the other hand however, using a VCO has some important advantages. First, the output clock jitter is only indirectly affected by the jitter of the reference signal, since the loop acts as a low-pass filter. Second, the output clock period can be a fraction of the reference clock period, if a frequency divider is implemented in the loop feedback path. This frequency multiplication property is the main reason for the widespread adoption of PLL's in applications such as microprocessor clock generation [20],[21],[22]. Moreover, since the VCO inherently generates a periodic clock signal, PLL's utilizing appropriate phase detector designs are commonly used in clock and data recovery applications [23].

Delay-Locked Loops on the other hand, make use of the fact that in many applications the reference signal is already a clock of the right frequency [24], [25]. Instead of generating their output clock with a VCO, DLL's use a Voltage Controlled Delay Line (VCDL) which generates the output clock by delaying its input clock by a controllable time delay. The phase of the VCDL output clock (*clk*) is compared by the phase detector with the phase of the reference clock. The output of the phase detector is filtered by the loop filter generating the control voltage, V_C. This control voltage drives the VCDL control input closing the negative feedback loop. Since the VCDL in this system is simply a delay gain element, the loop filter does not need a stabilizing zero and can be implemented by a single integrator (e.g. a charge pump and a capacitor). This control system is unconditionally stable resulting in a much easier design. In addition, a DLL can be easily implemented as a bang-bang control system, in which the phase detector output is simply a binary up-down phase error indication rather than a voltage proportional to the instantaneous phase error. In this case, the phase detector can be a replica of the input pin receiver. This way, the placement of the receiving clock edge compensates the setup-time of the input receiver, which is especially significant in guaranteeing timing margins at high bit-rates. In contrast, due to frequency acquisition constraints, PLL's usually rely on a state-machine-based phase-frequency detector, that results in sub-optimal placement of the receiving clock edge.

In the noisy environment of a digital IC, the most important difference between PLL's and DLL's is in the way they react to noise. The delay elements within the delay line or VCO have a sensitivity to supply or substrate noise. This performance measure is best expressed in a normalized units of percentage of delay change per percentage of supply or substrate change (%delay/%volt). The sensitivity varies considerably with the design of the elements. For example, a simple CMOS inverter has a supply sensitivity of 1-%delay/%supply; while, a well-designed differential buffer in Figure 12 has supply sensitivity of 0.2-%delay/%supply. Typically a PLL will have higher supply or substrate noise sensitivity than a DLL comprising identical delay elements [25]. Intuitively, a change on the supply or substrate of a VCO results in a change on its operating frequency. This frequency difference results in an increasing phase error that accumulates until the correcting action of the loop feedback takes effect. In contrast, the change

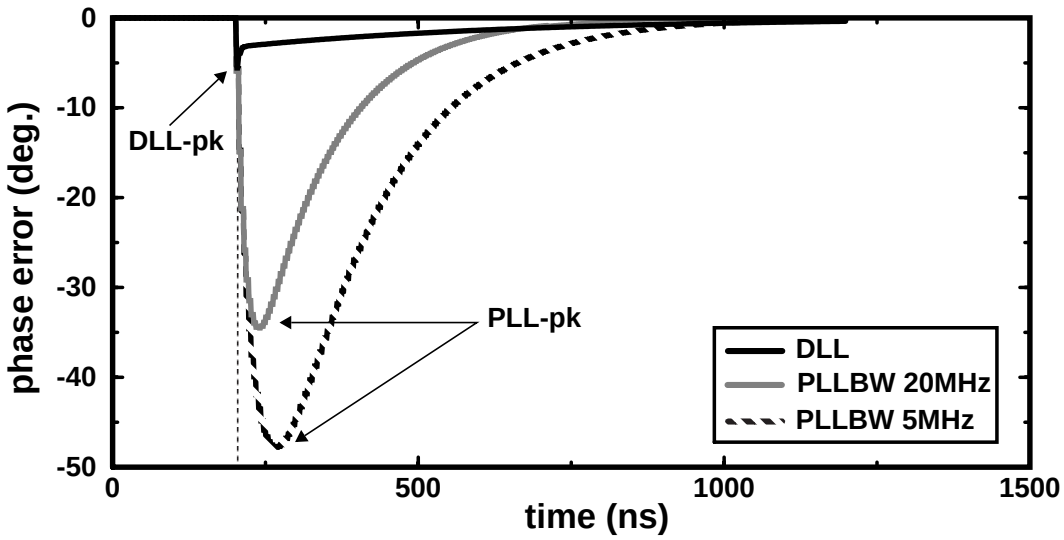


Figure 11: PLL and DLL transient response to supply step

on the supply of a VCDL results in a delay change through the delay line. Since the VCDL does not recirculate its output clock, the resulting phase error does not accumulate, and instead decreases with a rate proportional to the loop bandwidth. This performance difference is illustrated in Figure 11 which shows the simulated phase error of a PLL and a DLL under a supply voltage step. Both the PLL and the DLL are locked to a reference clock with a frequency of 250 MHz. The VCO and the VCDL comprise six voltage-controlled delay elements: each with a supply sensitivity of 1.8^{o}/Volt in a 3.3-V supply environment (i.e. a 1-Volt change in the supply of the VCO or VCDL with 4-ns cycle-time changes the delay through each element by 20-ps, corresponding to 0.2-%delay/%supply). A 300-mV supply step is applied on both the VCO and the VCDL 200-ns after the start of the simulation. As can be seen in Figure 11, the PLL peak phase error is generally larger than 6.5^{o} ($12x1.8^{o}x0.3$). The magnitude of this error depends both on the delay element supply sensitivity and the loop bandwidth. A larger loop bandwidth results in less phase error accumulation, thus minimizing the peak phase error. In contrast, the DLL phase error depends only on the supply sensitivity of the delay elements, and its peak occurs during the first clock cycle after the supply step. Even in the best case where the PLL bandwidth is 20-MHz the peak phase error is approximately a factor of 6 larger than that of the DLL (increasing the PLL bandwidth further than 1/10 of the operating clock frequency compromises the loop stability). Therefore, in applications such as high speed parallel signalling where no clock frequency multiplication is required, using a DLL maximizes the timing margins both because it exhibits lower supply and substrate induced phase noise, and because it can more readily use the input pin receiver as a phase detector.

It should however be noted, that other factors such as the quality of the system clock and the supply sensitivity of the final on-chip clock buffer can affect design trade-offs. For example, the above comparison does not include the supply sensitivity of the final on-chip clock buffer, which

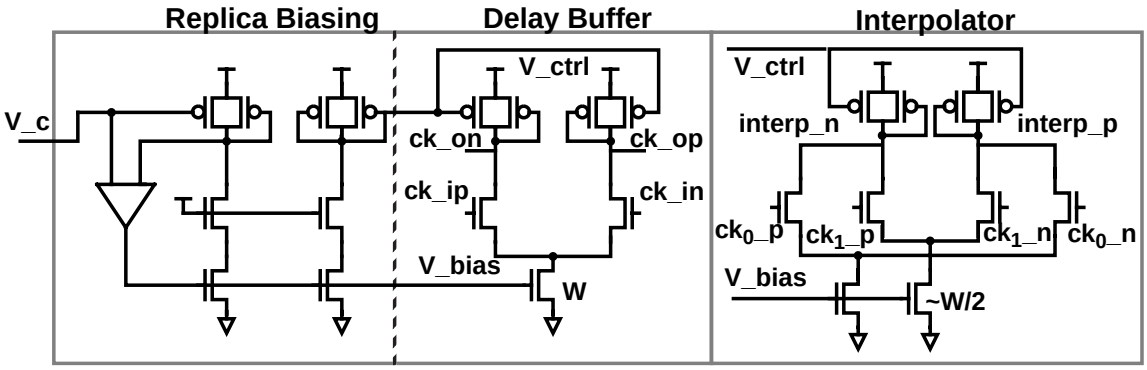


Figure 12: Delay element and interpolator design

typically comprises CMOS inverters. Because the supply sensitivity of an inverter is approximately 5x worse than that of the delay elements, a long buffer chain can contribute to a significant fraction of the total jitter. Thus the difference between a system utilizing a PLL and a system utilizing a DLL is often smaller than the factor of six quoted above. This smaller noise difference and the frequency multiplication capability of PLL's make them a better choice in many applications.

For the higher order multiplexing and de-multiplexing systems described earlier, precisely spaced clock phases are used to determine the bit-width. Several techniques can be used to generate these phases. The simplest is to use a ring oscillator (or a delay line with its phase input locked to its output phase) and tap the output of each of its stages. For example, a 6-stage oscillator employing differential stages, can generates 12 edges evenly spaced in the 0-360^{o} interval. An example of a robust differential delay element [26] with low supply sensitivity is shown in Figure 12. For even finer phase spacing than a single buffer delay, phase interpolators can be used to generate a clock edge occurring halfway between two input phases. The design of an interpolator uses two buffers -- each with different phases as inputs and each with a fraction of the drive strength. The sum of the drive strengths is equal to that of a normal buffer. The output is driven by the buffer with the earlier phase input for the period of the phase difference between the inputs before being driven by the full drive-strength. The resulting signal, compared to normally buffered versions of the inputs, has an intermediate phase which depends on the drive-strength ratio of the interpolating buffers. A design of the interpolator using the same type of buffer as the delay element is shown in Figure 12.[*] The quality of phase spacing generated by a ring oscillator and interpolators is measured to have errors of less than +/-8% of the ideal phase spacing. Alternative techniques for finely spaced phase generation, such as coupled oscillators and delay verniers, are described in [27],[29],[30].

Regardless of whether the clock generation employs a PLL or a DLL, a high loop tracking

[*] Because the interpolators are not perfect linear interpolators because they are not perfect integrators [7].

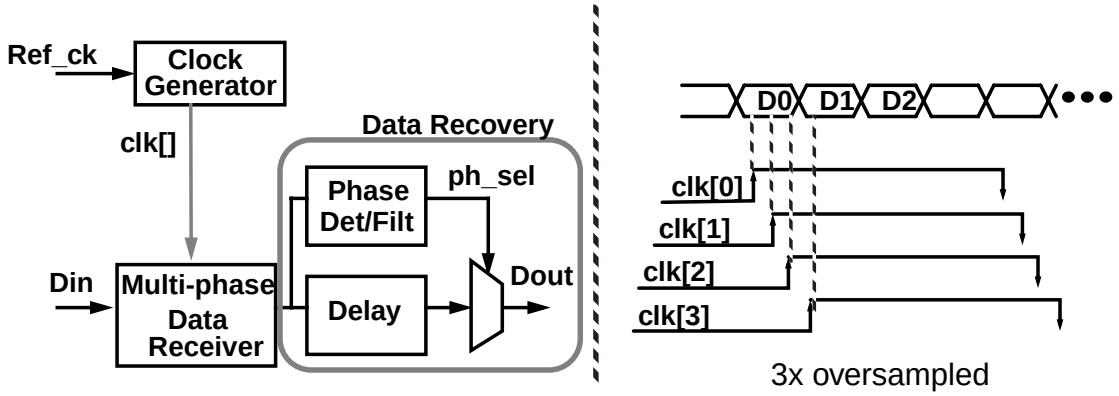


Figure 13: Phase-picking data recovery

bandwidth can reduce the effect of jitter. By tracking the phase variation of the data the system timing margin can be improved, provided that the phase variation is not a random cycle-to-cycle variation. An important area of current research is determining the actual on-chip noise spectrum. If the phase noise is correlated from one cycle to the next, a high tracking bandwidth is desired. Achieving a PLL bandwidth of the same order as the operating clock frequency is virtually impossible, because of loop stability constraints. A promising alternative used in UART's [28] is data oversampling. In this method, each data-bit from the data stream (*Din*) is sampled at multiple positions by multiple clock phases (*clk*). Data transitions are detected and the sample furthest away from the transition is picked (*ph_sel*). By delaying the samples while the decision is made, this method essentially employs a feed-forward loop (Figure 13), which due to the absence of stability constraints can achieve very high bandwidth and track phase movements on a cycle-to-cycle basis. However, the tracking can only occur at quantized steps depending on the degree of oversampling, and the phase picking decision incurs a large latency overhead.

B. Jitter and Phase Error Scaling

The basic speed of the clock scales with technology since the delay of each buffer will scale. Each buffer has a minimum delay less than a FO-4 inverter, so it is easy to use 4-6 buffers in a ring and generate the 8 FO-4 clock. Jitter and phase error should also scale with technology, if certain constraints are met. The supply/substrate sensitivity of a buffer is typically a constant percentage of its delay on the order of 0.1-0.3 %delay/%supply for differential designs with replica feedback biasing [26]. This implies that the supply and substrate induced jitter on a DLL scales with operating frequency -- the shorter delay in the chain, the smaller the phase noise. This argument also holds for the jitter caused by the buffer chain that follows the DLL. As technology scales, the delay of the buffer chain scales, and so does the resulting jitter. The jitter of a PLL scales with frequency if the loop bandwidth (and thus the input clock frequency) scales as well. If the reference clock does not scale due to system cost constraints, phase-picking can be effectively used to increase the tracking bandwidth and improve the robustness of the link at the expense of increased latency.

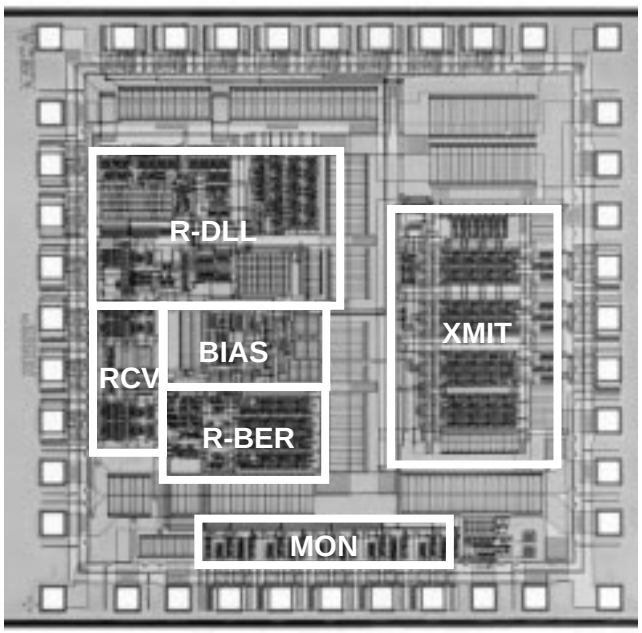


Figure 14: Low-latency 900-Mbps parallel link test chip die-photo

Imperfections in the static phase spacing would cause bit-width variations, thus reducing the aggregate data-eye. The errors are primarily introduced by mismatches in the delays of the ring oscillator stages and the corresponding buffering and interpolating paths. These delay mismatches can be caused by differences in coupling and load capacitance of the multiple parallel clock paths, as a result of layout matching errors. Since these errors can be minimized with careful design, a more important source of phase offset is the random mismatches between nominally identical devices such as transistor thresholds and widths. This source of timing errors becomes more severe with technology scaling, since the device threshold voltage becomes a larger fraction of the scaled voltage swing. Hence, phase spacing errors as a percentage of bit widths can be expected to increase with decreasing transistor feature sizes. However, the static nature of these errors will enable their cancellation by using static timing calibration schemes. For example, the interpolating clock generation architectures described in [7], [27] can be augmented with digitally controlled phase interpolators [31], and digital control logic to effectively cancel device induced phase errors. In addition to random device mismatches, decreasing bit-times in low latency parallel interfaces, will magnify the effect of mismatches on the electrical length of the parallel interconnects. Per channel timing adjustments can be used to mitigate this problem as well.

V. Link Performance Examples

Our research group built two different links that explore some of the design issues described in the previous sections. The first is a parallel, low latency (1.5 cycles) link, targeting a 4 FO-4 bit-time in a 0.8-µm process and employing pseudo-differential signalling. To improve the reception robustness, the chip uses a current integrating receiver described in detail in [32]. To minimize

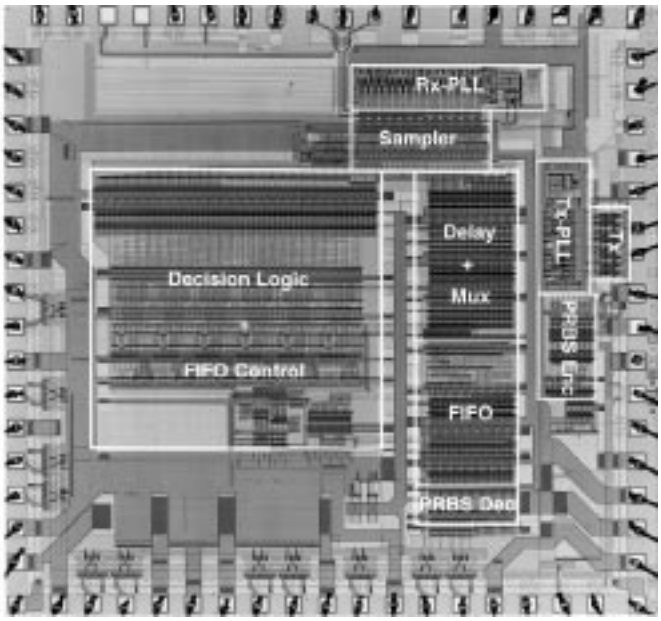


Figure 15: 4-Gbps high-speed serial link die photo

jitter a DLL comprising the delay elements in [26] was used. The chip, shown in Figure 14, includes PRBS testing and supply noise monitoring circuits. The design achieved a BER < 10^{-14} and a minimum bit-time of 3.3 FO-4 yielding a maximum transfer rate of 900-Mbps per parallel pin operating from a 4-V supply. The supply jitter sensitivity of the DLL was measured to be 0.7-ps/mV, 60% of which can be attributed to the final clock buffer.

A second chip [33] is a serial link transceiver targeting a data rate of 4-Gbps in a 0.6-µm (drawn) process with bit-width of ~1 FO-4. The chip uses an input regenerative amplifier based on the design in [11] as part of the 1:8 de-multiplexing receiver, and the 8:1 multiplexing transmitter of Figure 6-(b). Timing recovery is performed by a 3x oversampled phase-picking method similar to [6], [34]. . The latency of the data recovery is ~64 bits. The die photo in Figure 15 (3x3mm^2 chip) shows a considerable area penalty due to the phase picking decision logic. The chip achieved a of BER < 10^{-14} with the transmitter output fed back to the receiver operating at 3.3-V supply. A data eye at 3-Gbps (limited by the triggering bandwidth of the oscilloscope) is shown in Figure 16. The 330-ps data eye is closed by 90-ps due to the combined static phase spacing error and jitter. Due to the small delay of the on-chip clock buffers, the supply sensitivity of the clocking circuits is only 0.6-ps/mV.

VI. Channel Limitations

So far, the focus has been on the implementation and technology limitation of the transmit, receive, and timing recovery circuits without addressing the effects of the channel. We have shown that circuits should continue to scale with technology. Unfortunately the bandwidth of the

wires is not infinite, and will limit the bit-rate of simple binary signalling. A version of example parallel link circuits ported to a 0.25-µm technology achieved a 2-Gbps data rate [35]. The 4-Gbps rate of the serial link example is already higher than the bandwidth of copper cables longer than a few meters. While finite wire bandwidth is an issue that must be dealt with, it will not fundamentally limit signalling rate, at least not for a while. The question of how to maximize the number of bits communicated through a finite bandwidth channel is an old one. It is the basis of modem technology which push 30-50Kbps through the limited bandwidth (4KHz) of phone lines [36], [37], [38]. To counteract the channel limitations, complex schemes equalize the attenuation of the channel to extend the bandwidth, and more fully utilize the available bandwidth. These techniques are quickly reviewed in the following sections; see [16] for a more complete discussion. In this paper, we try to address the new issues caused by the intrinsic bandwidth of the medium, and the desire to keep the latency through the link small.

A. Cable Characteristics

The bandwidth limitation of the cable depends on its physical characteristics: the size and construction of their conductor and shield, and the dielectric material [44]. The thickness of the conductor (wire-gauge) determines the surface area in which the current can flow. Along with the conductivity of the metal, this area determines the conductor's resistance per meter. A signal current on the conductor requires a return path to close the circuit; the size of the loop and the proximity to the return path determines the inductance and capacitance. These characteristics determine the cable frequency response which is formally expressed by a transfer function. For better frequency response, cables are designed with a shield as the return path that is isolated from the signal by a fixed distance with a dielectric material. With this construction, the signal sees a

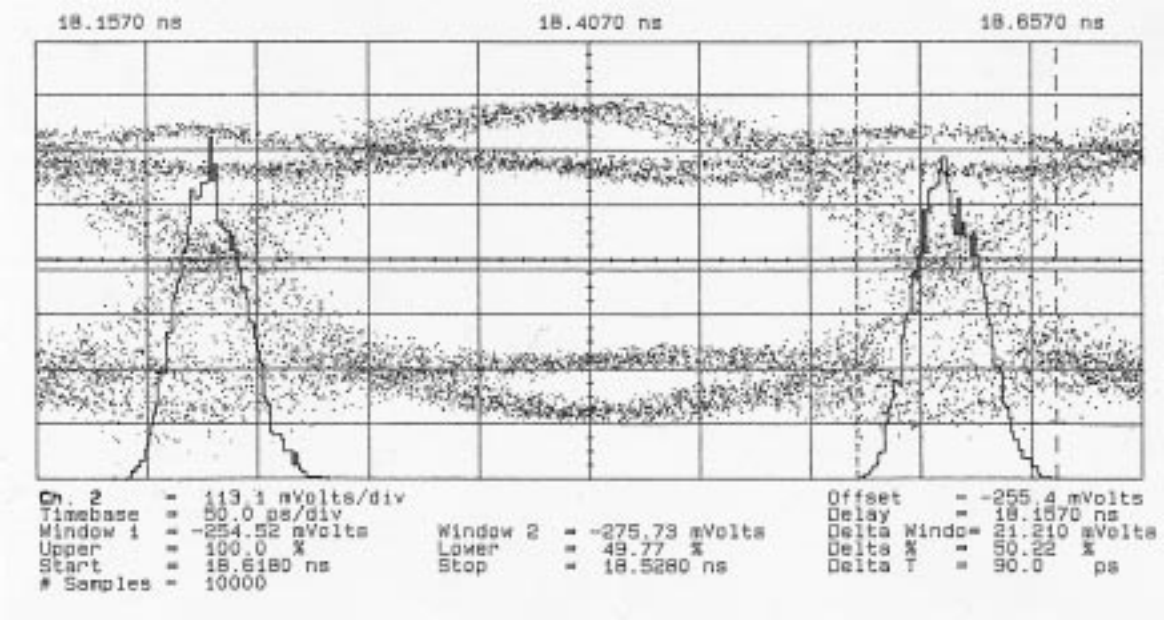


Figure 16: 3-Gbps data-eye from output of serial link multiplexing transmitter

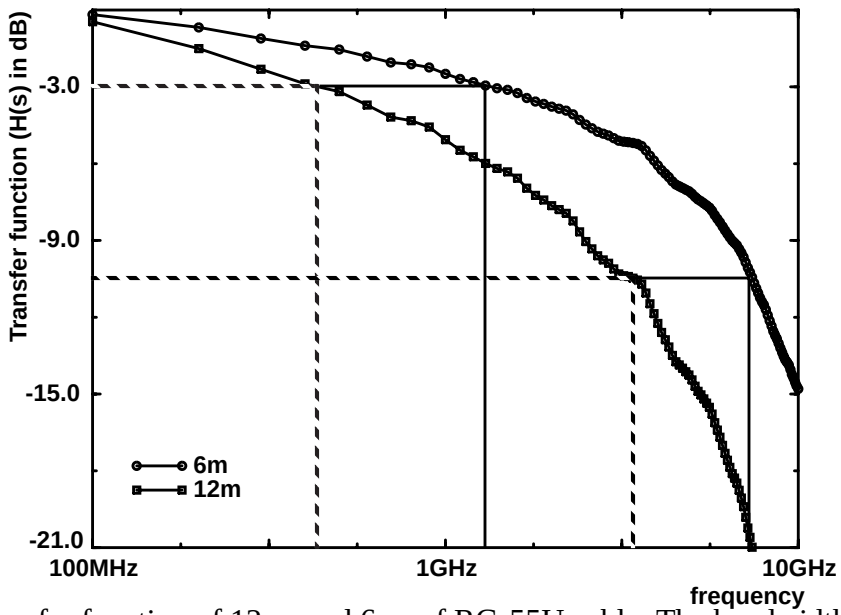


Figure 17: Transfer function of 12-m and 6-m of RG-55U cable. The bandwidth of the cable is usually set by the 3dB point (top line). By attenuating the low frequency components the effective bandwidth can be extended (bottom line)

distributed LC. The LC behaves as temporary energy storage that propagates a signal without loss down the cable as a wave. The effect of an ideal cable is just a delay of the signal depending on the cable length and no signal energy loss at any frequency. At the end of the line, a resistor with value $\sqrt{L/C}$ terminates the line and absorbs the propagated energy. In reality, such an ideal medium propagating all frequency components of the signal does not exist. The transfer function of a 6-m and 12-m RG55U cable, shown in Figure 17, illustrates increasing attenuation with frequency. The main source of the attenuation is the series resistance of the cable. In a phenomenon known as the skin effect, this resistance increases at higher frequencies because higher frequency currents travel closer to the conductor surface reducing the area of current flow. The increase in resistance is proportional to $\sqrt{frequency}$. Another, less important cause of attenuation is the loss of energy through the imperfect dielectric that isolates the signal from the shield.

Figure 18 demonstrates the time-domain effect of the frequency dependent attenuation. A single square pulse is injected into the 12-m cable. The attenuation above the pulse frequency causes the pulse amplitude to be reduced by more than 40%. Moreover, lower frequency attenuation results in the long settling tail exhibited by the signal. The net effect on a transmitted pseudo-random pattern is a significant closure of the resulting data eye, as illustrated in Figure 19. As a result, the bandwidth of the NRZ data is ultimately limited by the cable quality and length, unless extremely high quality cables are used which increases the overall system cost.

The simplest way to achieve higher data rate, even with the lossy channels, is to actively compensate for the uneven channel transfer function. This can be accomplished either through a

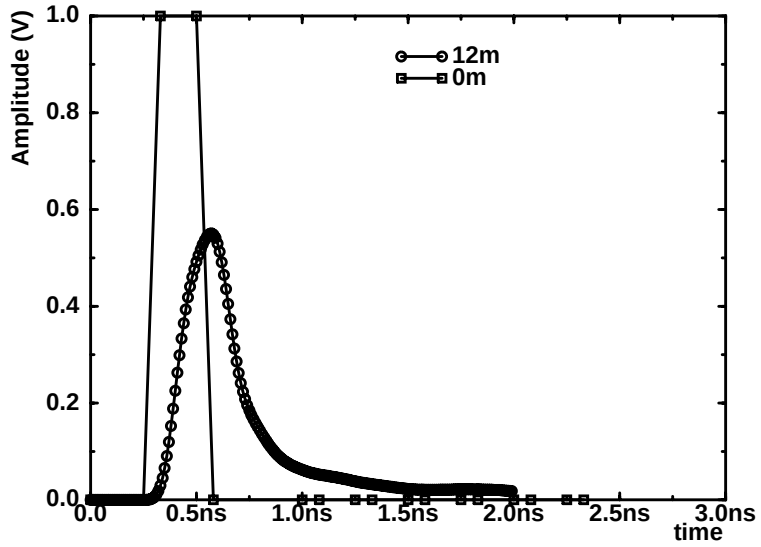


Figure 18: Time domain response of a square pulse into the 12-m RG-55U cable

pre-distorting transmitter [39],[40],[41],[42] or alternatively by an equalizing receiver [45]. The effect of both pre-distortion and equalization is similar: the system transfer function is multiplied by the pre-distorting/equalizing transfer function which in the ideal case is the inverse of channel transfer function. Unfortunately, this flattening of the channel frequency response comes at the expense of reduced signal to noise ratio (SNR). This is because equalization attenuates the lower frequency signal components so that they match the high frequency channel-attenuated components (shown in the horizontal lines in Figure 17). This reduces the overall signal power,

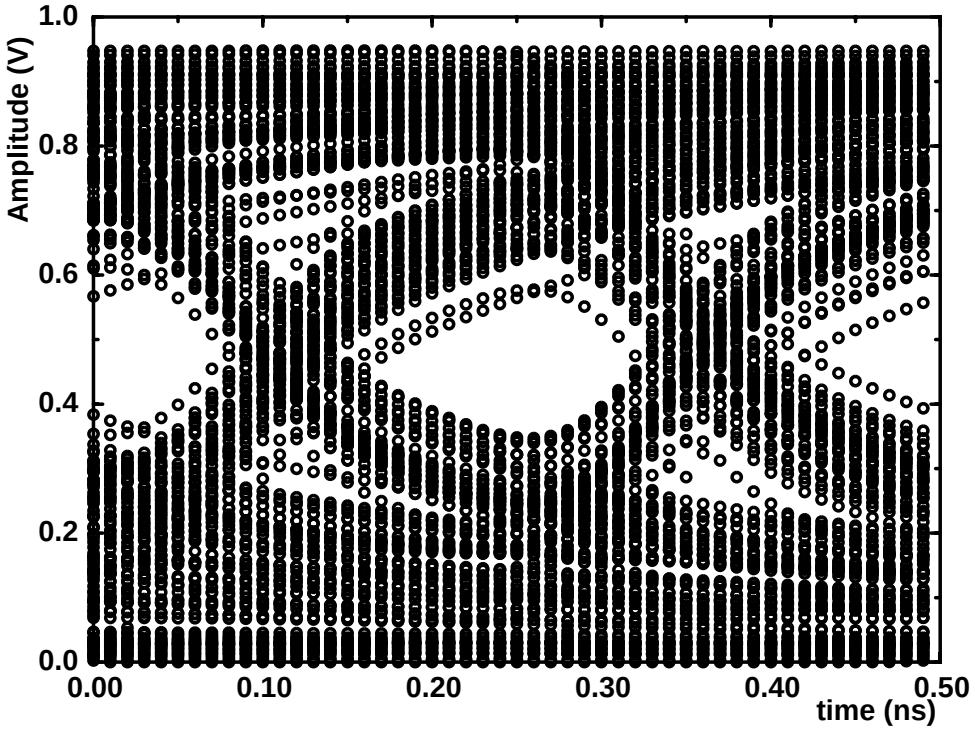


Figure 19: Simulated data-eye through a 12-m cable

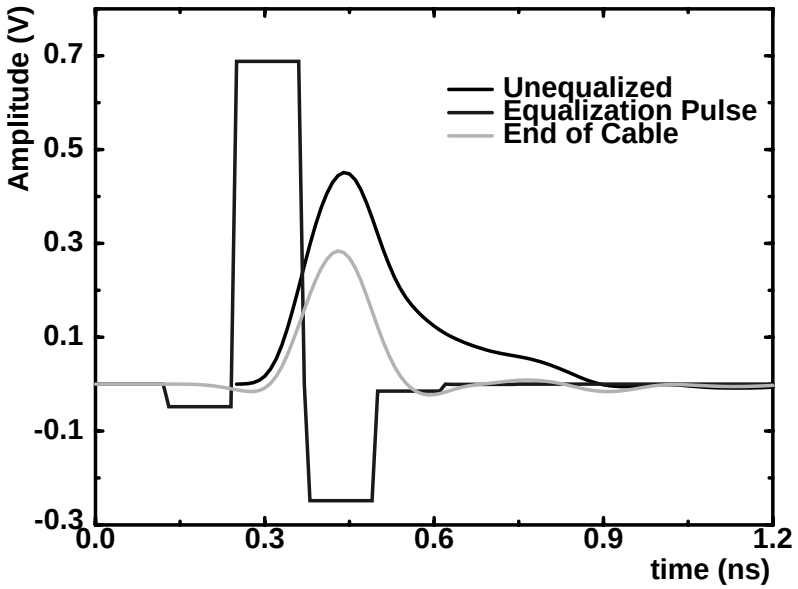


Figure 20: Pulse response from a pre-distorted transmitter

and hence degrades the system SNR. The advantage of these simple equalization techniques is that they can be implemented with little latency and power overhead.

B. Pre-distorting Transmitters

Transmitter pre-distortion utilizes a filter that precedes the actual line driver. The inputs to the filter are the current, past, and future bits that are being transmitted. The coefficients of the filter depend on the channel characteristics. The length of the optimal filter depends on the number of bits that affect the currently transmitted bit. Effectively, the FIR filter output is no longer a binary signal but a multi-level analog signal, while the output driver behaves as a high-frequency digital-to-analog converter (DAC) that operates at the bit-rate. In the simplest case, the FIR filter effectively suppresses the power of low frequency components by reducing the amplitude of continuous strings of same-value data on the line. Simultaneously it keeps the power of high signal frequency components the same by increasing the signal amplitude during transitions. Figure 20 shows the pre-distorted pulse, and the resulting pulse at the end of the cable as compared to the original pulse.

Because the transmitter does not have any information about the signal shape at the receiver-end of the channel, obtaining the appropriate FIR filter coefficients can be challenging. So far, pre-distorting high speed transmitters utilize static filter coefficients relying on a fixed channel transfer function. In the future, more intricate active training requiring back-channel information will probably be necessary in order to ensure robust operation under varying channel conditions.

C. Receiver Equalization

Receiver side equalization relies essentially on the same mechanism as pre-distortion, by utilizing a receiver with increased high frequency gain in order to flatten the system response [45]. This filtering can be implemented as an amplifier with the appropriate frequency response. Alternatively the filter can be implemented in the digital domain by feeding the input to an analog-to-digital converter (ADC) and post-processing the ADC's output with a high-pass filter. Similar to the pre-distorting transmitter, the overall system SNR is reduced, but because the high-pass filter also amplifies the noise at higher frequencies. Using an ADC at the input and building the filters digitally is the usual technique, since it also allows one to implement more complex and non-linear receive filters. While this approach works well when the data input is bandwidth limited to several MHz, it becomes more difficult with GHz signals which require Gsamples/s converters.

Equalization in either the transmitter or receiver is the simplest technique to use, and is effective for extending the bit-rate of the wires. Its cost is modest, especially if it can be done at the transmitter. However, equalization does not take full advantage of the channel, since it simply attenuates the low frequency signals resulting in smaller signal amplitudes. If SNR is large enough to detect these small signals, rather than attenuating all the signals, a more efficient scheme is to send many small signals at once (multi-bit signals) and detect them.

D. Multi-level Signalling

By transmit multiple bits in each transmit time, the required bandwidth of the channel for a given bit-rate is decreased. The simplest multi-level transmission scheme is pulse amplitude modulation. This requires a digital-to-analog converter (DAC) for the transmitter and an analog-to-digital converter (ADC) for the receiver. An example is 4 level pulse amplitude modulation (4-PAM) where each symbol-time comprises 2-bits of information. A 5 GSym/sec, 2-bit data-eye is shown in Figure 21 [43]. The pre-distortion and equalization discussed in the previous section can still be applied to improve the signal to noise ratio if the symbol-rate approaches or exceeds the channel bandwidth.

Shannon's Capacity Theorem determines the upper bound on maximum bit-rate. The equation, $C/f_{bw} = \log(1 + S/N)$, formulates the maximum capacity (bits/sec) per Hertz of channel bandwidth as a function of the signal to noise ratio (Figure 22). The larger the signal to noise ratio, the more information can be transferred for a given bandwidth by transmitting multi-level signals.

Once the receiver has to deal with multi-level signals, one can further improve the performance of the link by allowing symbols to overlap, and accounting for it in the detector. When the input consists of the sum of previous bits, a sequence detector (convolutional coders or Viterbi detectors [48]) is used to determine the transmitted bits. Instead of deciding the transmitted values on a bit-

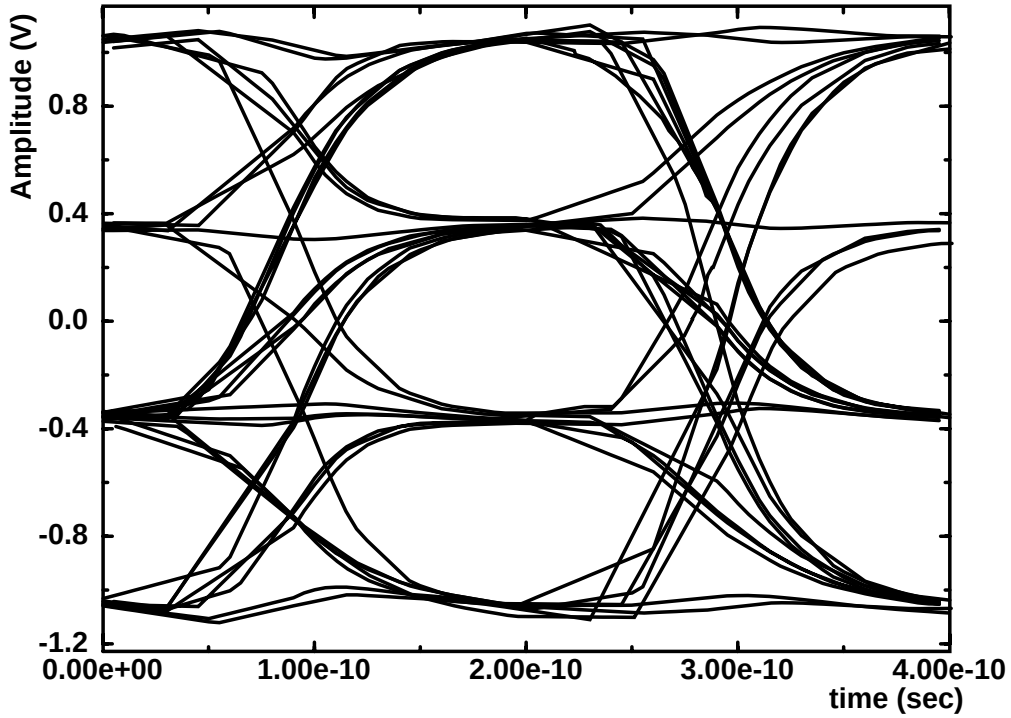


Figure 21: Simulated 4-PAM data-eye

by-bit basis, these detectors receive an entire bit sequence prior to making a decision. Thus they are able to use the signal energy that arrives late to help determine the value of the bit, and improve the effective SNR ratio. The downside of this technique is that the receiver input has effectively more values than the transmitted levels, thus requiring a higher resolution ADC.

E. DAC and ADC Requirements

All of the techniques discussed above require higher analog resolution both for the transmitter and

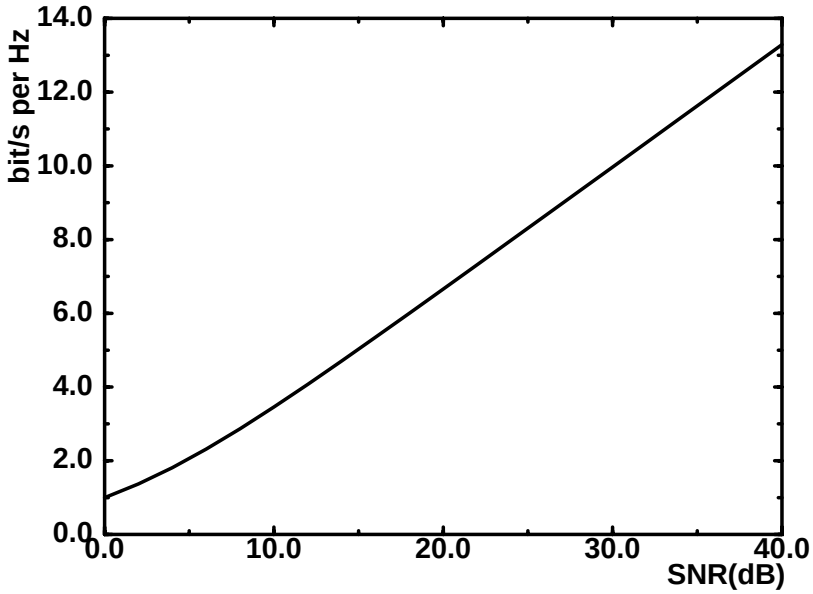


Figure 22: Shannon's Capacity Limit

the receiver. The trade-off for higher resolution transmitter DAC's and receiver ADC's is higher data throughput, and improved signal-to-noise ratio. The intrinsic noise floor for these converters is thermal noise ($\sqrt{4kTRf}$). In a 50Ω environment, the noise is ~1-nV/$\sqrt{Hz}$ (@ 4GHz, 3σ <200μV) which is negligible for 2-5 bits resolution even at GHz bandwidth.

Implementing high speed DAC's with less than 8-bit resolution is less challenging than implementing similar speed and resolution ADC's, because transmitter device sizes are large hence less sensitive to device mismatches. Instead, these DAC's are limited by circuit issues: primarily transmit clock jitter, and the settling of output transition induced glitches. Both of them are related to the intrinsic technology speed and therefore will scale with reduced transistor feature size. High-speed, low-resolution DAC's are rare; the nearest comparable product is found in video display RAMDAC's achieving 8-bit linearity operating at 320MHz on a 0.8-μm process [50].

The receive-side ADC is more difficult because it requires accurate sampling and amplification of signals on the order of 30-mV (5-bits of resolution for 1-V signal) at a very high rate. Such fast sampling requires a high sampling bandwidth (small sampling aperture) which depends on the on-resistance of the sampling FET(s) and the slew-rate of the clock driving the sample-and-hold network. Since both of these quantities scale well with technology, sampling aperture will not be a limiting factor of the ADC performance. To maintain good resolution, the aperture uncertainty must affect the sampled value by less than 1 LSB. For an ADC sampling a sine wave, the aperture uncertainty must be less than $1/(\pi f 2^m)$ [46] where m is the number of bits and f is the input frequency. The aperture uncertainty window has the same definition as described earlier dominated by jitter of the sampling clock, but it also includes an additional component from any non-linear voltage dependence of the sampling. Since jitter, as mentioned before, scales with technology and the sampling non-linearity can be characterized as a percentage error of the sampling aperture, the aperture uncertainty can also be expected to scale. The only factor not scaling well with technology is random device mismatch. Shrinking device dimensions decrease the total area over which the random mismatches can be averaged, thus increasing the total error [13]. In addition, in order to keep the receiver input capacitance low, smaller device sizes are required, increasing the effect of random device mismatch. Using active offset cancellation circuitry will become necessary to mitigate these effects [12], [46]. Examples of high speed ADC's can be found in disk-drive read-channels, where a 6-bit flash-ADC operating at 200MHz has been demonstrated in a 0.6-μm process technology [49]. Unfortunately this is still far below the 1-10GHz sample rate that a high-speed link will require. While there are no fundamental limitation preventing these devices, a significant amount of research needs to be done before they are pratical.

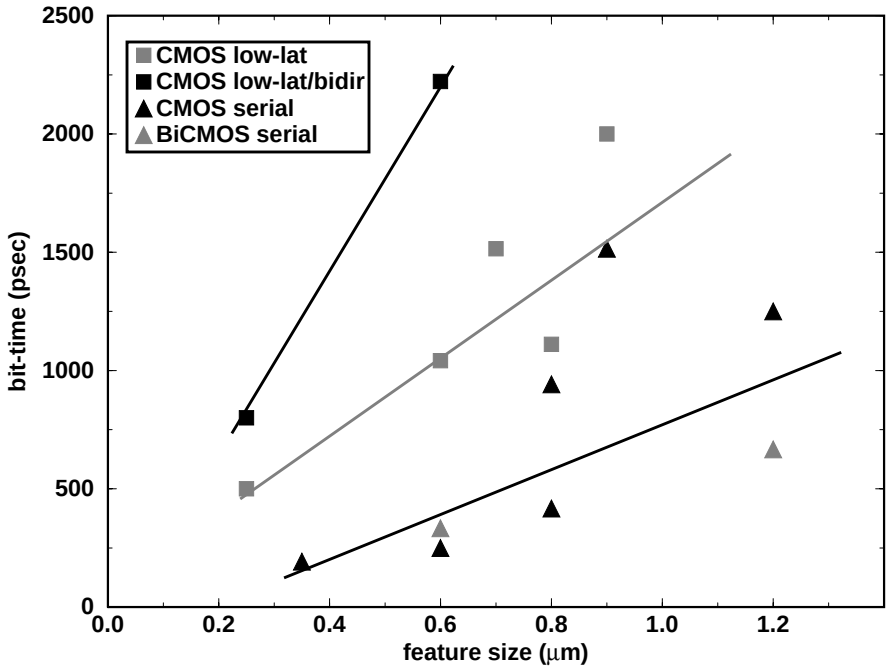


Figure 23: Link performance vs. technology for existing links

VII. Conclusion: Limits of Electrical Signalling

As technology scales, the capability of the electronics can be expected to scale along with the FO-4 metric. This scaling has been reflected in the progressively increasing bit-rates in NRZ signalling. Figure 23 shows the published performance for both serial and parallel links fabricated in different technologies. Low latency parallel links have maintained bit-times of ~3-4 FO-4 inverter delays. Meanwhile, by applying high degrees of parallelism, the off-chip data rate of serial links has exceeded on-chip limitations, achieving bit-times of ~1 FO-4 inverter delays. To track technology scaling, additional complexity is required to handle the increased device mismatches and on-chip noise levels. Although jitter is expected to scale with technology, more attention must be paid to the clocking architecture, in order to minimize phase noise and phase offset of the on chip clock and maintain high bit-rates.

The fundamental limitation of bit-rate scaling stems from the communication channel, rather than the circuit fabrication technology. Frequency dependent channel attenuation limits the bandwidth, by introducing inter-symbol interference. As long as sufficient signal power exists, the simplest technique to extend the bit-rate beyond the cable bandwidth is transmitter pre-distortion. Similar but slightly more complex is receive-side equalization. Also with sufficient signal power, more bits can be transmitted per symbol by employing various modulation techniques. If the total signal power is limited, complex techniques such as sequence detection and coding can be applied to best utilize the available bandwidth. As Shannon's Theorem predicts, the limit on the bits/sec of information that can be transmitted per Hz of channel bandwidth depends entirely on how much signal power can be expended.

The cost of applying these methods is more digital signal processing, longer latency, and higher analog resolution. Digital processing complexity is a strength of CMOS technology and therefore it does not impose a limitation, other than the required increase in die area and power consumption. Much of the processing can be pipelined and parallelized to meet speed requirements. Also, as technology improves, the processing ability scales. However, the longer latency from applying these more elaborate techniques will limit their application to longer serial links. For low-latency intra-system communication, only simpler techniques such as transmitter pre-distortion can be applied without significant latency penalty. Lastly, higher analog resolution is a critical requirement in both applications. These techniques have been applicable to lower rates primarily because high quality (low jitter and phase offset) clocks along with high resolution DAC's and ADC's are available. Current research efforts are concentrated in low-resolution and high sampling bandwidth DA/AD converters, and timing offset cancellation both for serial and parallel links. Besides Shannon's Limit, there is no fundamental limit to signalling rates, though many implementation challenges certainly exist.

VIII. References

[1] Wickelgren, I.J. “The facts about FireWire [serial communication bus]” *IEEE Spectrum*, April 1997. vol.34, no.4, p. 19-25

[2] Thompson, G.O. “Work progresses on gigabit Ethernet” *Computer IEEE Comput. Soc*, May 1997. vol.30, no.5, p. 95-6

[3] Kushiyama, N. et al. “A 500-megabyte/s data rate 4.5 M DRAM” *IEEE Journal of Solid-State Circuits*, April 1993. vol.28, no.4, p. 490-8

[4] Mooney, R. et al. “A 900 Mb/s bidirectional signalling scheme” *IEEE Journal of Solid-State Circuits*, Dec. 1995. vol.30, no.12, p. 1538-43

[5] Takahashi, T. et. al. “A CMOS Gate Array with 600 Mb/s Simultaneous Bidirectional I/O Circuits,” *IEEE Journal of Solid State Circuits*, vol. 30, no. 12. Dec 1995

[6] Kyeongho Lee et al. “A CMOS serial link for fully duplexed data communication”, *IEEE Journal of Solid-State Circuits*, April 1995. vol.30, no.4, p. 353-64

[7] Yang, C.K.K. et al. “A 0.8-µm CMOS 2.5 Gb/s oversampling receiver and transmitter for serial links”, *IEEE Journal of Solid-State Circuits*, vol.31, no.12 p. 2015-23

[8] Kim, B. et al. “A 30-MHz hybrid analog/digital clock recovery circuit in 2-µm CMOS” *IEEE Journal of Solid-State Circuits*, Dec. 1990. vol.25, no.6, p. 1385-94.

[9] Hu, T.H. et al. “A monolithic 480 Mb/s parallel AGC/decision/clock-recovery circuit in 1.2-µm CMOS”, *IEEE Journal of Solid-State Circuits*, Dec. 1993. vol.28, no.12, p. 1314-20

[10] Donnelly, K.S. et al. “A 660 MB/s interface megacell portable circuit in 0.3µm-0.7µm CMOS ASIC”, *IEEE Journal of Solid-State Circuits*, Dec. 1996, vol.31, no.12 p. 1995-2003

[11] Yukawa, A. et al. “A CMOS 8-bit high speed A/D converter IC”. Proceedings of the *Tenth European Solid-State Circuits Conference* p. 193-6

[12] Jieh-Tsorng Wu et al. “A 100-MHz pipelined CMOS comparator” *IEEE Journal of Solid-State Circuits*, vol.23, no.6, p. 1379-85

[13] Pelgrom, M.J. “Matching Properties of MOS Transistors,” *IEEE Journal of Solid State Circuits*, vol. 24, no. 10 pp. 1433, Dec. 1989

[14] Sidiropoulos, S. et al. “A CMOS 500-Mbps/pin synchronous point to point link interface” *Proceedings of 1994 IEEE Symposium on VLSI Circuits. Digest of Technical Papers* p. 43-4

[15] Galles, M. et al. “Spider: a high-speed network interconnect” *IEEE Micro* Jan.-Feb. 1997. vol.17, no.1, p. 34-9

[16] Proakis, J, Salehi M., *Communication Systems Engineering,* Prentice Hall 1994

[17] Gardner, F.M. “Charge-pump phase-lock loops” *IEEE Transactions on Communications,* Nov. 1980. vol.COM-28, no.11, p. 1849-58

[18] Best, R., “Phase-Locked Loops”, 3rd ed, McGraw Hill, 1997

[19] Gardner, F. “Phase Lock Techniques”, Wiley 1979

[20] Jeong, D.K. et al. “Design of PLL-based clock generation circuits” *IEEE Journal of Solid-State Circuits*, April 1987. vol.SC-22, no.2, p. 255-61

[21] Young, I.A. et al. “A PLL clock generator with 5 to 110 MHz lock range for microprocessors” *IEEE Journal of Solid-State Circuits*, Nov. 1992. vol.27, no.11, p. 1599-607

[22] Alvarez, J. et al. “A wide-bandwidth low-voltage PLL for PowerPC microprocessors” *IEEE Journal of Solid-State Circuits*, April 1995. vol.30, no.4, p. 383-91

[23] Razavi, B. Editor, “Monolithic Phase Locked Loops and Clock Recovery Circuits”, IEEE Press, 1996.

[24] Johnson, M.G. et al. “A variable delay line PLL for CPU-coprocessor synchronization” *IEEE Journal of Solid-State Circuits*, Oct. 1988. vol.23, no.5, p. 1218-23

[25] Lee, T.H. et al. “A 2.5 V CMOS delay-locked loop for 18 Mbit, 500 megabyte/s DRAM” *IEEE Journal of Solid-State Circuits*, Dec. 1994. vol.29, no.12, p. 1491-6

[26] Maneatis, J.G. et al. “Low-jitter process-independent DLL and PLL based on self-biased techniques” *IEEE Journal of Solid-State Circuits*, Nov. 1996. vol.31, no.11, p. 1723-32

[27] Maneatis, J.G. et al. “Precise delay generation using coupled oscillators” *IEEE Journal of Solid-State Circuits*, Dec. 1993. vol.28, no.12, p. 1273-82

[28] McNamara, J. E. *Technical Aspects of Data Communication*, 2nd ed. Digital Press, Bedford, MA, 1982.

[29] Moyer, G.C. et al. “The delay vernier pattern generation technique” *IEEE Journal of Solid-State Circuits,* April 1997. vol.32, no.4, p. 551-62

[30] Gray, C. et al. “A Sampling Technique and Its CMOS Implementation with 1Gb/s Bandwidth and 25-ps Resolution”, *IEEE J. of Solid State Circuits*, vol. 29, no. 3, ‘94

[31] Sidiropoulos, S. et al. “A semi-digital DLL with unlimited phase shift capability and 0.08-400 MHz operating range” *1997 IEEE International Solids-State Circuits Conference, Digest of Technical Papers,* p. 332-3.

[32] Sidiropoulos, S. et al. “A 700-Mb/s/pin CMOS signalling interface using current integrating receivers” *IEEE Journal of Solid-State Circuits,* May 1997. vol.32, no.5, p. 681-90

[33] Yang, C.K.K. et al. “A 0.6-µm CMOS 4-Gbps Transceiver with Data Recovery using Oversampling” *Proceedings of 1997 IEEE Symposium on VLSI Circuits. Digest of Technical Papers* p. 71-72

[34] Kim, S. et al. “An 800Mbps Multi-Channel CMOS Serial Link with 3x Oversampling,” *IEEE 1995 CICC Proceedings,* pp. 451, Feb. 1995.

[35] Chang, K.Y. et. al. “A 2Gb/s Asymmetric Link for High-Bandwidth Packet Switches”, *Hot Interconnects V Symposium Record*, August 1997, Stanford, 171-179

[36] Ishida, H. et al. “A single-chip V.32 bis modem”, *1994 IEEE International Solid-State Circuits Conference. Digest of Technical Papers.* p. 66-7

[37] Combe, M. et al. “A second-generation modem analog front-end”, *1992 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*. p. 224-5, 290

[38] Boxho, J. et al. “An analog front end for multi-standard power line carrier modem” *1997 IEEE International Solid-State Circuits Conference. Digest of Technical Papers.* p. 84-5

[39] Widmer, A.X. et al. “Single-chip 4*500-MBd CMOS transceiver”, *IEEE Journal of Solid-State Circuits,* vol.31, no.12 p. 2004-14, Dec. 1996

[40] Fiedler, A. et al. “A 1.0625 Gbps transceiver with 2x-oversampling and transmit signal pre-emphasis”, *1997 IEEE International Solids-State Circuits Conference, Digest of Technical Papers* p. 238-9

[41] Dally, W.J. et al. “Transmitter equalization for 4-Gbps signalling” *IEEE Micro,* Jan.-Feb. 1997. vol.17, no.1, p. 48-56

[42] Poulton, J. et. al. “A Tracking Clock Recovery Receiver for 4Gb/s Signalling”, *Hot Interconnects V Symposium Record*, August 1997, Stanford, 157-169

[43] Farjad-Rad, R. et al. “An Equalization Scheme for 10Gb/s 4-PAM Signalling over Long Cables” *Mixed Signal Conference*, July 1997, Cancun, Mexico.

[44] Matick, Richard “Transmission lines for digital and communication networks” 3rd ed., IEEE Press, 1997

[45] Bang-Sup Song et al. “NRZ timing recovery technique for band limited channels” 1996 *IEEE International Solid-State Circuits Conference. Digest of Technical Papers,* p. 194-5.

[46] Van de Plassche, R. *Integrated Analog-to-Digital and Digital-to-Analog Converters*, Kluwer Academic Publishers, 1994.

[47] International C.C.I.T.T. Recommendations, “V.29-V.33”, Geneva

[48] Black, P.J. et al. “A 1-Gb/s, four-state, sliding block Viterbi decoder” *IEEE Journal of Solid-State Circuits*, June 1997. vol.32, no.6, p. 797-805

[49] Spalding, J. et al. “A 200M sample/s 6b flash ADC in 0.6μm CMOS” 1996 *IEEE International Solid-State Circuits Conference. Digest of Technical Papers,* p. 320-1.

[50] Reynolds, D. “A 320 MHz CMOS Triple 8 bit DAC with PLL” *IEEE Journal of Solid-State Circuits*, Dec. 1994. vol 29, no.12, pp. 1545